

IN THE CLAIMS

Please cancel claims 1-7 and 24-33.

Please add new claims 34-55 as indicated below.

1-33 (Cancelled).

34. (New) A system for accessing a region of memory in a computer, the system comprising:

a network interconnect;

a first node coupled to the interconnect, wherein the first node includes:

a memory configured to store data;

a memory region table;

a memory window table, said memory window table being in

communication with the memory region table; and

circuitry configured to control access to the memory;

a second node coupled to the first node via the interconnect, wherein the second node is configured to convey a request for access to the memory in the first node, the request including a first virtual address and a remote access key;

wherein in response to receiving the request, the first node is configured to:

determine whether the request corresponds to a memory region or a memory window;

if the request corresponds to a memory region:

combine the first virtual address and the remote access key to form an index into a first entry of the memory region table; and

if the request corresponds to a memory window:

- (i) use a first portion of the remote access key to index into a first entry of the memory window table;
- (ii) compare a second portion of the remote access key to a window table access key stored within the first entry of the memory window table; and
- (iii) allow processing to continue if the second portion of the remote access key matches the window table access key.

35. (New) The system as recited in claim 34, wherein the first node is configured to determine whether the request corresponds to a memory region or a memory window by examining a most significant bit of the remote access key.

36. (New) The system as recited in claim 35, wherein the first entry of the memory window table further includes a new remote access key, and wherein the system is further configured to combine the new remote access key with the first virtual address to form an index into a second entry into the memory region table, in response to determining the second portion of the remote access key matches the window table access key.

37. (New) The system as recited in claim 36, wherein the request includes a first length, and wherein the first entry of the memory window comprises a second virtual address and second length, wherein the first node is configured to combine the new remote access key with the first virtual address in further response to determining the first virtual address and first length are contained within the address space defined by the second virtual address and the second length.

38. (New) The system are recited in claim 35, wherein the first entry of the memory region table includes a first protection domain value, and wherein the first node is further configured to:

compare the first protection domain value with a second protection domain value of a
send work queue which initiated the request;
allow access if the first protection domain value matches the second protection
domain value; and
deny access if the first protection domain value does not match the second protection
domain value.

39. (New) The system as recited in claim 39, wherein in response to allowing access, the first node is further configured to utilize a physical address included within the first entry of the memory region table to access the memory.

40. (New) The system as recited in claim 39, wherein the second entry of the memory region table includes a second protection domain value, and wherein the first node is further configured to:

compare the second protection domain value with a third protection domain value of a
send work queue which initiated the request;
allow access if the second protection domain value matches the third protection
domain value; and
deny access if the second protection domain value does not match the third protection
domain value;
wherein in response to allowing access, the first node is further configured to utilize a
physical address included within the second entry of the memory region table
to access the memory.

41. (New) The system as recited in claim 40, wherein the memory region table and the memory window table reside on one or more modules external to and in communication with an operating system.

42. (New) The system as recited in claim 40, wherein an entry of the memory window table comprises:

a memory region record; and
a memory window record.

43. (New) A method for controlling access to a memory in a computing node, said method comprising:

receiving a request for access to the memory, the request including a first virtual address and a remote access key;
determining whether the request corresponds to a memory region or a memory window by examining a bit included within the request;
in response to determining the request corresponds to a memory region,
combining the virtual address and the remote access key to form an index into a first entry of a memory region table, in response to determining the request corresponds to a memory region; and
in response to determining the request corresponds to a memory window, (i) utilizing a first portion of the remote access key to index into a first entry of a memory window table, (ii) comparing a second portion of said remote access key to a window table access key stored within the first entry of the memory window table, and (iii) allowing processing to continue if the second portion of the remote access key matches the window table access key.

44. (New) The method as recited in claim 43, wherein said determining comprises examining a most significant bit of the remote access key.

45. (New) The method as recited in claim 44, wherein the first entry of the memory window table further includes a new remote access key, and wherein the method further comprises combining the new remote access key with the first virtual address to form an index into a second entry into the memory region table, in response to determining the second portion of the remote access key matches the window table access key.

46. (New) The method as recited in claim 45, wherein the request includes a first length, and wherein the first entry of the memory window comprises a second virtual address and second length, wherein the method further comprises combining the new remote access key with the first virtual address in further response to determining the first virtual address and first length are contained within the address space defined by the second virtual address and the second length.

47. (New) The method are recited in claim 44, wherein the first entry of the memory region table includes a first protection domain value, and wherein the method further comprises:

- comparing the first protection domain value with a second protection domain value of a send work queue which initiated the request;
- allowing access if the first protection domain value matches the second protection domain value; and
- denying access if the first protection domain value does not match the second protection domain value.

48. (New) The method as recited in claim 47, wherein in response to allowing access, the method further comprises utilizing a physical address included within the first entry of the memory region table to access the memory.

49. (New) The method as recited in claim 47, wherein the second entry of the memory region table includes a second protection domain value, and wherein the method further comprises:

- comparing the second protection domain value with a third protection domain value of a send work queue which initiated the request;
- allowing access if the second protection domain value matches the third protection domain value; and
- denying access if the second protection domain value does not match the third protection domain value;

wherein in response to allowing access, the method further comprises utilizing a physical address included within the second entry of the memory region table to access the memory.

50. (New) A node for use in a system area network, the node comprising:

an adapter configured to receive a memory access request via a network interconnect, said request including a virtual memory address and a remote access key;

a memory configured to store data; and

circuitry configured to control accesses to said memory, wherein in response to receiving a memory access request, said circuitry is configured to:
determine whether the request corresponds to a memory region or a memory window, responsive to examining a bit included within the request;

if the request corresponds to a memory region:

combine the first virtual address and the remote access key to form an index into a first entry of a memory region table; and

if the request corresponds to a memory window:

(i) use a first portion of the remote access key to index into a first entry of a memory window table;

(ii) compare a second portion of the remote access key to a window table access key stored within the first entry of the memory window table; and

(iii) allow processing to continue if the second portion of the remote access key matches the window table access key.

51. (New) The node as recited in claim 50, wherein the circuitry is configured to determine whether the request corresponds to a memory region or a memory window by examining a most significant bit of the remote access key.

52. (New) The node as recited in claim 51, wherein the first entry of the memory window table further includes a new remote access key, and wherein the circuitry is further configured to combine the new remote access key with the first virtual address to form an index into a second entry into the memory region table, in response to determining the second portion of the remote access key matches the window table access key.

53. (New) The node as recited in claim 52, wherein the request includes a first length, and wherein the first entry of the memory window comprises a second virtual address and second length, wherein the circuitry is configured to combine the new remote access key with the first virtual address in further response to determining the first virtual address and first length are contained within the address space defined by the second virtual address and the second length.

54. (New) The node are recited in claim 52, wherein the first entry of the memory region table includes a first protection domain value, and wherein the circuitry is further configured to:

- compare the first protection domain value with a second protection domain value of a send work queue which initiated the request;
- allow access if the first protection domain value matches the second protection domain value; and
- deny access if the first protection domain value does not match the second protection domain value.

55. (New) The node as recited in claim 54, wherein in response to allowing access, the circuitry is further configured to utilize a physical address included within the first entry of the memory region table to access the memory.